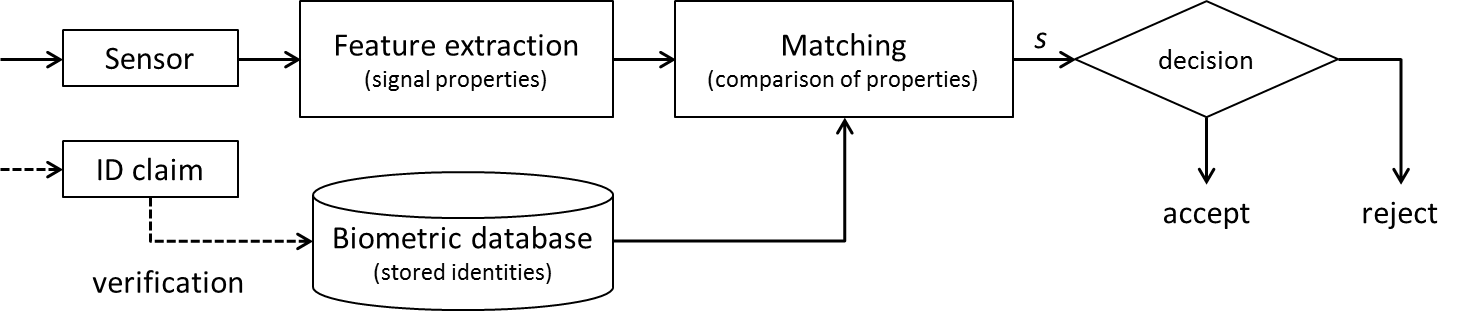
# Hand geometry.

Each person has unique biometric characteristics that can be used to automatically determine the person's identity. Examples of such biometric characteristics are: fingerprints, face, iris, voice and hand geometry.

A system for automatic person recognition, called biometric system, typically consists of several modules, as shown in the figure:



**Figure 1**. Biometric system. In identification mode (one-to-many comparison), only the sensor data is used to determine the person's identity. In verification mode (one-to-one comparison), the person must also claim a target identity. A decision regarding the person's identity is made based on the value of the similarity measure between the sensor data and the database data.

**Hand geometry as a biometric identifier.**

An image of the hand can be obtained for example with a scanner (see Figure 2).



**Figure 2**. Hand image scanned with a RICOH Aficio 3235 scanner. Some examples of measurements used to describe the hand are shown. These measurements are collected in a vector called feature vector which is the description of the hand geometry.

The length and width of the fingers, palm, etc. are measured from the hand image. These dimensions (hand properties) can be used to describe the hand geometry. Some examples of such measures are indicated in Figure 2. The measurements are collected in a *feature vector* E=[e(1), e(2), ..., e(n)] which represents a new description of the hand (besides the hand image itself). The feature vector is used by the automated system to identify the person.

Although the hand dimensions may not vary so much between individuals, the hand geometry can be used to verify the identity of a person with sufficient accuracy for “convenience” applications (if you want to know more about this, read the article: " *A Prototype Hand Geometry-based Verification System* " which is included with this exercise).

**Database creation.**

A feature vector for each person *authorized* to use the recognition system needs to be added to the biometric database (this special feature vector, called *reference vector* or *enrolment vector*, will be stored in the database, and used as reference for each person *authorized* to use the system). A reference vector can be created for example by taking the average of each hand measurement for a number of several feature vectors of the same hand. Importantly, these feature vectors are calculated from hand images captured at different times (different scans).

Each reference vector represents a point in a multidimensional space (feature space). The position of these points in the feature space depends on the geometry of the hand (measurements). Points with similar characteristics (e.g. from images of the same hand) are ideally grouped together in the same region of the feature space, while points with different characteristics (e.g. from images of different hands) should appear separated in different regions of the feature space

**Comparison of characteristics.**

As already mentioned, hands with similar characteristics point closer together in the feature space, while hands with different characteristics point to different regions of the feature space. This means that the similarity between two feature vectors can be calculated using distance measures.

In ***identification*** mode, the feature vector of the input image is compared with all reference vectors stored in the database. These distances are ranked (ordered) in increasing order, and a list of associated identities is created. The identity corresponding to the minimum distance appears first in the list, and so on. The “winner” is set as the identity with the minimum distance.

In ***verification*** mode, only one distance is calculated: the distance between the feature vector of the input image, and the reference vector corresponding to the claimed identity. The distance is compared against a predefined threshold; if the distance is less than the threshold, the identity is verified (i.e. accepted), otherwise it is rejected.

Example:

Three persons of reference R1, R2 and R3 (person 1, 2 and 3) are enrolled into the system database. Each reference vector contains two dimensions r(1) and r(2), which is a measure of the person's height and weight. This means that the feature space is two-dimensional.

A person X wants to use the system, so the height e(1) and weight e(2) is used to build the feature vector E = [e (1), e (2)].

Some example values:

Let´s assume the feature vectors: R1=[175, 80], R2=[192, 85], R3=[160, 58] and E=[176, 82].

**In identification mode,** recognition is done using distance di between the feature vector E and the reference vectors Ri (i = 1,2,3) via:

, which is the Euclidean distance between points

We thus calculate the distances:

 (distance between E and R1)

 (distance between E and R2)

 (distance between E and R3)

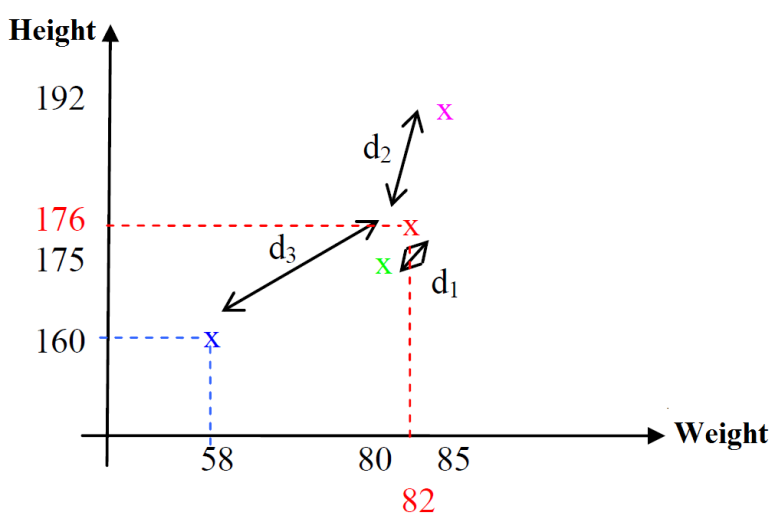
Distances are then ranked (ordered) in increasing order and the associated identities are put in a list L, in our example L = {R1, R2, R3}. The first identity in the list is the “winner”, so the person X is identified as R1 (person 1)

It is also possible to add to the "winner" the condition that the distance must be sufficiently small (e.g. d<5). This is to ensure that the winning reference identity is sufficiently close to the person X. The value of this distance (here = 5) is called *threshold*.

**In verification mode**, the person X would claim that his/her identity is person 1. Therefore, in this case we only calculate distance  which is then compared against a threshold value (e.g. threshold = 5). Since , we verify the identity of person X as being person 1.

Graphically, this example can be seen as follows (see figure):

The properties height and weight of the reference identities R1, R2 and R3 and the feature vector E correspond to points of the feature space (marked with x in the figure). The elements of each vector represent the coordinates, so the position in the feature space is determined by the values of the vector, e.g. x = (160,58) for R3.



**Figure 3.** 2-dimensional feature space. Reference vectors R1=[175,80], R2=[192,85] , R3=[160,58] and the feature vector E=[176,82] of person X are marked as points in the feature space. Distances between reference vectors R1, R2, R3 and the feature vector E are represented with arrows. Person X is identified as person 1 since d1 is the minimum distance among d1, d2 and d3.

# EXERCISE.

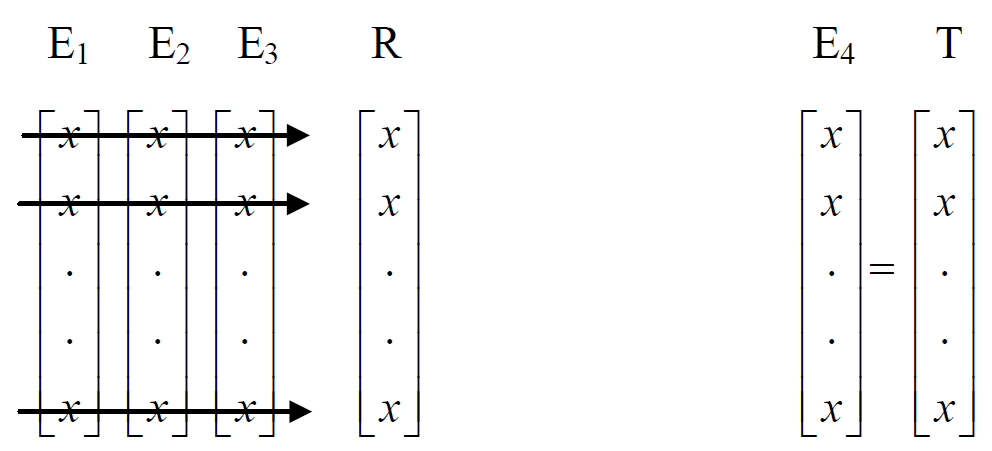
**Measure hand properties.**

Measure in each image provided, the hand geometry properties (finger’s length and width, palm width, etc.) according to the "measuring template" attached with the exercise. Use a ruler and measure in mm, e.g. 20 mm. You will then obtain four feature vectors E1, E2, E3 and E4, one for each image.

*NOTE! It is important that you try to measure the properties always in the same way in each image. This can be done if you make use of the natural lines that are in your hand (this is important, if you do not measure the properties consistently in the same manner, you are reducing the chances of the system to identify you successfully).*

Compute the reference vector R of your own identity by using three of the four feature vectors. For this purpose, calculate the average of each characteristic in these three feature vectors. Make use of one decimal point for average, e.g. 20.3 mm (see Figure 4).

The fourth feature vector will be used in the identification / verification tasks as test vector T.



**Figure 4**. Three of the feature vectors are used to form the reference vector R. The arrow in the figure should be interpreted as the mean value of the three characteristics in E1, E2, E3. The average value is placed in the reference vector R. The fourth feature vector is used as test vector T in the identification / verification tasks.

# Create database matrix and the test matrix.

Create a database matrix “DB” of reference vectors, and a matrix “TEST” of test vectors. For this purpose, we will use your vectors, and the vectors attached as .mat files with this exercise.

Each column in the matrix DB consists of a reference vector of a different person, and the column index is the ID number of the person to whom the reference vector belongs.

Each column in the matrix TEST is a test vector. The column index is also the ID number of the person to whom the test vector belongs.

*NOTE! A person must have the same column index in the DB and TEST matrices.*

Put the reference vector R of your own hand images in the first column of DB, and then concatenate the matrix of the file DB11a.mat in the following columns. Similarly, concatenate your test vector T with the matrix of the file T11a.mat. You should then obtain two matrices of 9x32 elements, with your ID corresponding to the first column.

***Save your DB and TEST matrices, and do not forget to attach them with the report of this exercise.***

Example: how to create matrices in Matlab.

Test the following Matlab commands to create matrices M1, M2 and M.

M1 = [1 2 3; 4 5 6; 7 8 9]';

M2 = [10 11 12; 13 14 15]';

M = [M1, M2]; % Add M1 and M2 consecutively

**Identification mode.**

The attached Matlab function **MinDistClassID(**'**eucl**'**,DB,TEST,id)** carries out the identification of the persons specified in the input parameter “id”. Input parameter 'eucl' indicates that the Euclidean distance should be used in the comparison. Input parameters DB and TEST are the database data and test data.

Input parameter “id” can be a single ID number or a set of more than one numbers.

For example:

**id=5;** %identify person with ID number 5

**id=[1 2 3 4 5];** % identify people with ID numbers 1,2,3,4 and 5

**id=[3 5 8];** % identify people with ID numbers 3, 5 and 8

**id=[1:30];** % identify people with ID numbers 1-30

Example of identification of person with ID number 5:

**load DB;** %load database data

**load TEST;** %load test data

**id=5;** %ID number

**[IdLista, IdAvst]=MinDistClassID('eucl',DB,TEST,id);** %perform identification

and example of identification of persons with ID numbers 1-30:

**id=[1:30];** % ID numbers

**[IdLista,IdAvst]=MinDistClassID('eucl',DB,TEST,id);** % perform identification

Output parameters IdLista and IdAvst contains the results of the identification, a column for each identification (one-to-many comparison), sorted by distance with the smallest distance first. IdLista contains column index of the reference vector (= identity) and IdAvst contains the distance between the test vector and the reference vector.

*Before you proceed:*

Carry out an identification with id = your ID number. Study results of the identification given by the function *MinDistClassID*, i.e. study the output parameters IdLista and IdAvst so you can interpret how the results are presented (if you cannot interpret the result of the identification, ask the instructor).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Identification exercise*

*Carry out identification for the whole group, i.e. for all people in the test data matrix TEST. Then use the results to:*

*1) Calculate by hand the function P (k) for k = 1, 2, ....*

*P (k) indicates the chance (probability) that the correct identity is output in position k of IdLista. Draw the function P(k) that you calculated in a graph.*

*2) Calculate by hand the function TPIR (M) and draw it in a graph (the CMC curve).*

*TPIR (M) indicates the chance (probability) that the correct identity is output in the first M positions of the output list. TPIR (M) is obtained by summing up P(k) up to k = M.*

*Put also the values of P (k) and TPIR (M) in a table.*

*3) From TPIR (M), calculate the length of the list M to have a 90% chance that the correct identity is obtained in IdLista among the first M identities.*

*4) Find the two ID numbers that are the most and the least similar to your own ID. Also write down the distances of the two cases (how similar the hands are).*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**Verification mode.**

To perform an authentication (one-to-one comparison) it is necessary to determine a value of the distance between the test vector and the reference vector which can be accepted as "sufficiently similar". This value is often called *threshold*.

The attached Matlab function **MinDistClassVER('eucl',DB,TEST,id,Th)** carries out the verification of the persons specified in the input parameter “id” using the threshold “Th”.

Example of verification of person with ID number 5:

**load DB;** %load database data

**load TEST;** %load test data

**id=5;** %ID number

%id=1:30; % example for ID numbers 1-30

**Th=.....;** %selection of a suitable threshold

**[VerId, VerAvst]=MinDistClassVER('eucl',DB,TEST,id,Th);** %perform verification

The output parameter VerId is a column vector. The value in the column vector may be 0 or 1, and interpreted as 0 = "acceptance" and 1 = "reject". Output parameter VerAvst is the distance between the test and reference vector.

*Before you proceed:*

Carry out a verification with id = your ID number. Study results of the verification given by the function *MinDistClassVER*, i.e. study the output parameters VerId and VerAvst so you can interpret how the results are presented (if you cannot interpret the result of the verification, ask the instructor).

In order to estimate the error of the system, we should compute statistics of the output distances when hands from the same person and from different persons are compared. This can be done with the provided function: **[ShDist,OhDist]=DistNew('eucl',DB,TEST).**

Input parameters DB and TEST are the database data and test data.

Output parameter ShDist contains the distances when hands from the same person are compared (ID=1 of DB with ID=1 of TEST, then ID=2 of DB with ID=2 of TEST, and so on)

Output parameter OhDist contains the distances when hands from different persons are compared (ID=1 of DB with ID=2,3,4… of TEST, then ID=2 of DB with ID=1,3,4… of TEST, etc.)

The histograms of the same-hand comparisons (Genuine class) and different-hands comparisons (Impostor class) are calculated and plotted with:

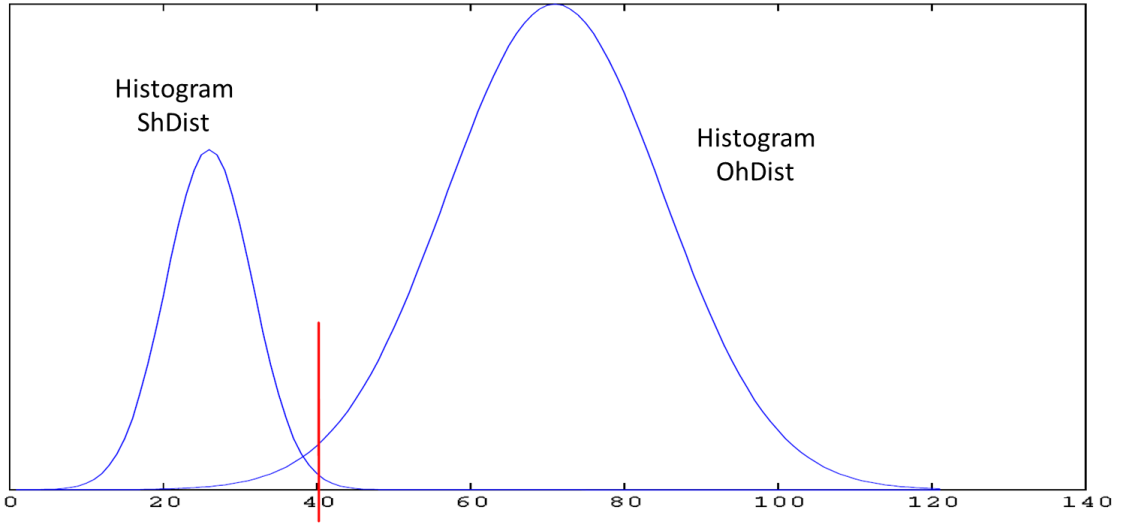
**[G,indG]=hist(ShDist,20);** % Calculate histogram class Genuine

**[I,indI]=hist(OhDist,50)**; % Calculate histogram class Imposter

**figure(100);bar(indG,G)**; % Plot histograms

**figure(101);bar(indI,I);**

From these histograms, errors are estimated for different thresholds. How? For a schematic description, see legend of Figure 5 below.



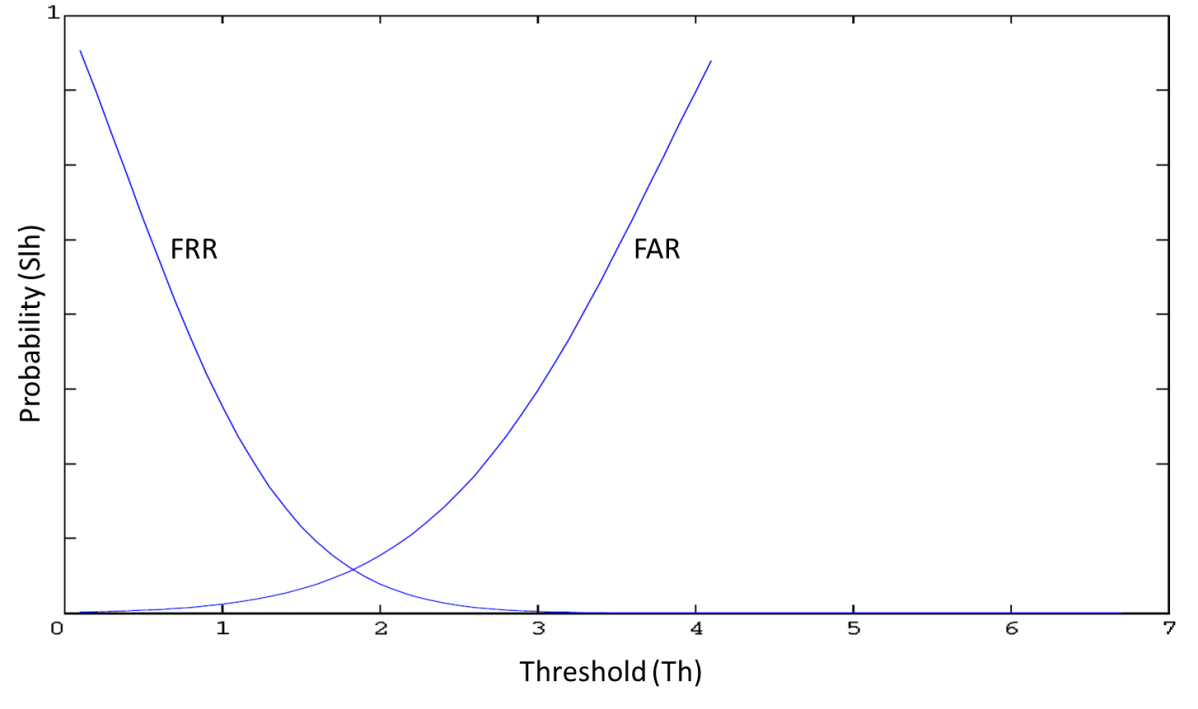
*distance (d)*

**Figure 5**. Example of histograms of ShDist and OhDist (the y-axis represents the number of elements and the x-axis represents distance). OhDist values (Impostor class) will mostly be larger than ShDist values (Genuine class), and this difference can be used to verify a person's identity by setting an appropriate threshold that allows to separate the two classes. Example of a threshold value is 40, which is marked with a red line. The area of the OhDist curve to the left of the red line represents the FA error, and the area of the ShDist curve to the right of the red line represents the FR error.

The threshold value (Th) is used to determine the limit for "sufficiently similar" between the test vector and the reference vector when carrying out verification (one-to-one comparison). If the calculated distance d between the two vectors under comparison is larger than the threshold value, then the person’s identity is not verified (= it is rejected).

Two errors can occur when doing verification, called FR (False Rejection) and FA (False Acceptance). An FR error occurs when a person claims his/her true ID, but the identity is not verified (= it is rejected) because d>Th. An FA error occurs when a person claims a wrong ID, but the identity is verified (= it is accepted) because d<Th.

The probability (Slh) of the two errors as a function of the threshold value can be plotted as shown in Figure 6 below. FRR stands for False Rejection Rate and FAR for False Acceptance Rate.



**Figure 6**. Example of how error can be presented for verification (one-to-one comparison). Slh for FR-error (= FRR) decreases as the threshold increases (= requirement to be less similar) but at the same time the FA-error (= FAR) increases when Slh increases (= impostor comparisons allowed to be less similar). Reducing the threshold (= requiring to be more similar) will produce the contrary effect, the FRR will increase while the FAR will decrease.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*Verification exercise*

*With the help of the histograms you created earlier and shown in Figure = 100 (Genuine class) and figure = 101 (Impostor class), calculate and plot the probability of the FA error (= FAR) and FR error (= FRR) for various thresholds. You can also work directly with the distances in the vectors ShDist (Genuine) and OhDist (Impostor). Create a table with different thresholds and the number of errors of each type.*

*A Matlab command that you can use to calculate the number of FR errors at a specific threshold Th is:* ***sum (ShDist> Th)****. Vary the value of Th and annotate both Th and the number of FR errors in a table.*

*How can you compute the probability (Slh) from the number of FR errors, i.e. FRR? Calculate FRR for each value of Th and annotate the FRR in the table. Now you can plot the FRR for different values of the threshold Th! Use Matlab function plot for this purpose.*

*Create a similar table and graph of the class Impostor. Which is the Matlab command to compute FA errors using OhDist and Th?*

*Present the estimated errors (FRR and FAR for different values of Th) of the system in two tables and in a figure as Figure 6.*

*From your results, indicate the approximate value of the EER (Equal Error Rate). Suggest also suitable values of Th for: 1) high security requirement, and 2) high convenience requirement. Indicate the value of Th, FAR and FRR for every case 1) and 2).*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# SUBMISSION OF THE REPORT.

Submit the report of this exercise no later than one week after the second lab session. Additional instructions are given in Blackboard.

Do not forget to attach your DB and TEST matrices (as .mat files) including your measured hand properties, so the teacher can replicate your results if necessary.